10/531969 #2

Rec'd PCT/PTO 20 APR 2005

PCT/IB03/04520

14 OCT 2003

Europäisches Patentamt

European Patent Office

Office européen des brevets

REC'D 20 OCT 2003

WIPO PCT

Bescheinigung

Certificate

Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. Patent application No. Demande de brevet n°

02292624.0

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

Der Präsident des Europäischen Patentamts; Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets p.o.

R C van Dijk

BEST AVAILABLE COPY

Anmeldung Nr:
Application no.: 02292624.0
Demande no:

Anmeldetag:
Date of filing: 23.10.02
Date de dépôt:

Anmelder/Applicant(s)/Demandeur(s):

Koninklijke Philips Electronics N.V.
Groenewoudseweg 1
5621 BA Eindhoven
PAYS-BAS

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.
If no title is shown please refer to the description.
Si aucun titre n'est indiqué se referer à la description.)

Method of building a variable-length error code

In Anspruch genommene Prioriät(en) / Priority(ies) claimed /Priorité(s) revendiquée(s)
Staat/Tag/Aktenzeichen/State/Date/File no./Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation/International Patent Classification/
Classification internationale des brevets:

H03M13/00

Am Anmeldetag benannte Vertragstaaten/Contracting states designated at date of filing/Etats contractants désignées lors du dépôt:

AT BE BG CH CY CZ DE DK EE ES FI FR GB GR IE IT LI LU MC NL PT SE SK TR

"METHOD OF BUILDING A VARIABLE-LENGTH ERROR CODE"

FIELD OF THE INVENTION

The present invention relates to a method of building a variable length error code and to a corresponding device.

BACKGROUND OF THE INVENTION

A classical communication chain, illustrated in Fig.1, comprises a source encoder followed by a channel encoder and, after the transmission of the coded signals thus obtained, a channel decoder and a source decoder. Variable-length codes (VLC) are classically used in source coding for their compression capabilities, and the associated channel coding techniques combat the effects of the real transmission channel (such as fading, noise, etc.). However, since source coding is intended to remove redundancy and channel coding to re-introduce it, it has been investigated how to efficiently coordinate these techniques in order to improve the overall system while keeping the complexity at an acceptable level.

Among the solutions proposed in such an approach, the variable-length error correcting (VLEC) codes present the advantage to be variable-length while providing error correction capabilities, but building these codes is rather time consuming for short alphabets (and become even prohibitive for higher length alphabets sources), and the construction complexity is also a drawback, as it will be seen.

First, some definitions and properties of the classical VLC must be recalled. A code C is a set of S codewords $\{c_1, c_2, c_3, \ldots, c_S\}$, for each of which a length $\ell_i = |c_i|$ is defined, with $\ell_1 \leq \ell_2 \leq \ell_3 \leq \ldots \leq \ell_i \leq \ldots \leq \ell_S$ without any loss of generality. The number of different codeword lengths in the code C is called R, with $R \leq S$, and these lengths are denoted as $L_1, L_2, L_3, \ldots, L_i, \ldots L_R$, with $L_1 < L_2 < L_3 < \ldots < L_R$. A variable-length code, or VLC, is then the structure denoted by (s_1@ L_1, s_2@ L_2, s_3@ L_3, ..., s_R@ L_R), which corresponds to s_1 codewords of length L_1, s_2 codewords of length L_2, s_3 codewords of length L_3, ..., and s_R codewords of length L_R. When using a VLC, the compression efficiency, for a given source, is related to the number of bits necessary to transmit symbols from said source, and the measure used to estimate this efficiency is often the average length AL of the code (i.e. the average number of bits needed to transmit a word), given, when each symbol a_i is mapped to the codeword c_i, by the following relation (1) :

$$AL = \sum_{i=1}^{i=S} \ell_i . P(a_i) \qquad (1)$$

which is equivalent to the relation (2) :

$$AL = \sum_{i=1}^{R} L_i \,.\, \left(\sum_{j=r(i)+1}^{j=r(i+1)} P(a_j) \right) \qquad (2)$$

where, for a data source A, the s source symbols are denoted by $\{a_1, a_2, a_3, \ldots, a_s\}$ and $P(a_i)$ is the respective probability of occurrence of each of these symbols, with $\Sigma\, P(a_i) = 1$ (from $i = 1$ to $i = s$). If Al_{min} denotes the minimal value for the average length AL, it is easy to see that when Al_{min} is reached, the symbols are indexed in such a way that $P(a_1) \geq P(a_2) \geq P(a_3) \geq \ldots \geq P(a_s)$. In order to encode the data in such a way that the receiver can decode the coded information, the VLC must satisfy the following properties : to be non-singular (all the codewords are distinct, no more than one source symbol being therefore allocated to one codeword) and to be uniquely decodable (*i.e.* it is possible to map any string of codewords to the correct source symbols string without any error).

An introduction and a presentation of different distances will then help to recall the notion of error-correcting property used in the VLEC codes theory :

(a) Hamming weight and distance : if w is a word of length n with $w = (w_1, w_2, \ldots, w_n)$, the Hamming weight of w is the number W(w) of non-zero symbols in w :

$$W(w) = \sum_{i=1}^{i=n} \frac{w_i}{\|w_i\|} \qquad (3)$$

and, if w_1 and w_2 are two words of equal length with $w_i = (w_{i1}, w_{i2}, w_{i3}, \ldots, w_{in})$ and $i = 1$ or 2, the Hamming distance between w_1 and w_2 is the number of positions in which w_1 and w_2 differ :

$$H(w_1, w_2) = W(w_1 - w_2) \qquad (4)$$

However, the Hamming distance is by definition restricted to fixed-length codes, and other definitions will be defined before considering VLEC codes.

(b) let $f_i = w_1^i\ w_2^i \ldots w_n^i$ be a concatenation of n words of a VLEC code C, then the set $F_N = \{f_i : |f_i| = N\}$ is called the extended code of C of order N.

(c) minimum block distance and overall minimum block distance : the minimum block distance b_k associated to the codeword length L_k of a VLEC code C is defined as the minimum Hamming distance between all distinct codewords of C with length L_k :

$$b_k = \min \{H(c_i, c_j) : c_i, c_j \in C, i \neq j, |c_i| = |c_j| = L_k\} \text{ for } k = 1, \ldots, R \qquad (5)$$

and the overall minimum block distance b_{min} of said VLEC code C, which is the minimum block distance value for every possible length L_k, is defined by :

$$b_{min} = \min \{b_k : k = 1, \ldots R\} \qquad (6)$$

(d) diverging distance and minimum diverging distance : the diverging distance between two codewords of different length $c_i = x_{i_1} x_{i_2} \ldots x_{i_{\ell_i}}$ and $c_j = x_{j_1} x_{j_2} \ldots x_{j_{\ell_j}}$ of a VLEC code C, where $|c_i| = \ell_i > |c_j| = \ell_j$, is defined by :

$$D(c_i, c_j) = H(x_{i_1} x_{i_2} \ldots x_{i_{\ell_i}}, x_{j_1} x_{j_2} \ldots x_{j_{\ell_j}}) \qquad (7)$$

i.e. it is also the Hamming distance between a ℓ_j - length codeword and the ℓ_j - length prefix of a longer codeword, and the minimum diverging distance d_{min} of said VLEC code C is the minimum value for all diverging distances between every possible couple of codewords of C of different length :

$$d_{min} = \min \{ D(c_i, c_j) : c_i, c_j \in C, |c_i| \neq |c_j| \} \qquad (8)$$

(e) converging distance and minimum converging distance : the converging distance between two codewords of different length $c_i = x_{i_1} x_{i_2} \ldots x_{i_{\ell_i}}$ and $c_j = x_{j1} x_{j2} \ldots x_{j_{\ell_j}}$ of a VLEC code C, where $|c_i| = \ell_i > |c_j| = \ell_j$, is defined by :

$$C(c_i, c_j) = H (x_{i_{\ell_i - \ell_j + 1}} x_{i_{\ell_i - \ell_j + 2}} \ldots x_{i_{\ell_i}}, x_{j_1} x_{j_2} \ldots x_{j_{\ell_j}}) \qquad (9)$$

i.e. it is also the Hamming distance between a ℓ_j - length codeword and the ℓ_j - length suffix of a longer codeword, and the minimum converging distance of said VLEC code C is the minimum value for all converging distances between every possible couple of C of different length :

$$c_{min} = \min \{ C(c_i, c_j) : c_i, c_j \in C, |c_i| \neq |c_j| \} \qquad (10)$$

(f) free distance : the free distance d_{free} of a code is the minimum Hamming distance in the set of all arbitrary extended codes :

$$d_{free} = \min \{ H(f_i, f_j) : f_i, f_j \in F_N, N = 1, 2, \ldots, \infty \} \quad (11)$$

Following the structure model used for a VLC, it is therefore possible to describe the structure of the VLEC code C by the notation :

$$S_1 @ L_1, b_1 ; S_2 @ L_2, b_2 ; \ldots ; S_R @ L_R, b_R ; d_{min}, c_{min} \qquad (12)$$

where there are s_i codewords of length L_i with minimum block distance b_i, for all i = 1, 2,... R, and minimum diverging and converging distances d_{min} and c_{min}. The most important parameter of a VLEC code is its free distance d_{free}, which influences greatly its performance in terms of error-correcting capabilities, and it can be shown that the free distance of a VLEC code is bounded by :

$$d_{free} \geq \min (b_{min}, d_{min} + c_{min}) \qquad (13)$$

These definitions being recalled, the state-of-the-art in VLEC codes construction will be now described more easily. The first type of VLEC codes, called α - prompt codes and introduced in 1974, and an extension of this family, called $\alpha_{t_1, t_2, \ldots, t_R}$ - prompt codes, have both the same essential property : if one denotes by $\alpha(c_i)$ the set of words that are closer to c_i than to any codeword cj, with $j \neq i$, no sequence in $\alpha(c_i)$ is a prefix of a sequence in another $\alpha(c_j)$. The construction of these codes is very simple, and the construction algorithm is adjustable by the number of codewords at each length, which makes possible to find the best prompt code for a given source and a given d_{free}. However, this best code performs poorly in terms of compression performance.

A more recent construction, allowing the construction of a VLEC code from the generator matrix of a fixed-length linear block code, was proposed in the document "Variable-length error-correcting codes" by V.Buttigieg, Ph.D.Thesis, University of Manchester, England, 1995. Called code-anticode construction, this algorithm relies on line combinations and column permutations to form an anticode at the rightmost column. Once the code-anticode generator matrix is obtained, the VLEC code is simply obtained by a matrix multiplication.

This technique has however several drawbacks. First, there is no explicit method to find the needed line combinations and column permutations to obtain the anticode. Moreover, the construction does not take into account the source statistics and, consequently, often reveals itself sub-optimal (one can find a code with smaller average length by a post-processing on the VLEC code). In the same document, the author has then proposed an improved method, called Heuristic method, that is based on a computer search for building a VLEC code giving the better known compression rate for a specified source and a given protection against errors, i.e. a code C with specified overall minimum block, diverging and converging distances and with codeword lengths matched to the source statistics so as to obtain minimal average length for the chosen free distance (in practice, one takes $b_{min} = d_{min} + c_{min} = d_{free}$ and $d_{min} = [d_{free}/2]$.

The main steps of this Heuristic method, which uses the following parameters : minimum length L_1 of codewords, maximum length L_{max} of codewords, free distance d_{free} between each codeword, number s of codewords required, are now described with reference to the flowcharts of Figs.2 to 4.

To start the computer search ("Start"), all the needed parameters must be first specified : L_1 (the minimal codeword length), L_{max} (the maximum codeword length), the different distances between codewords (d_{free}, b_{min}, d_{min}, c_{min}), and s (the number of codewords required by the given source), and some relations are set when choosing these parameters :

$$L_1 \geq d_{min}$$
$$b_{min} = d_{free}$$
$$d_{min} + c_{min} = d_{free}$$

The first phase of the algorithm, referenced 11, is then performed : it consists in the generation of a fixed length code (put initially in C) of length L_1 and minimal distance b_{min}. This phase is in fact an initialization, performed for instance by means of an algorithm such as the greedy algorithm (GA), presented in Fig.5, or the majority voting algorithm (MVA), presented in Fig.7, or a new proposed variation, denoted by GAS (Greedy Algorithm by Step), which consists in a variation of the two above mentioned ones. The GAS consists in the search method used in the GA, where instead of deleting half of the codewords, only the last codeword of the group is deleted. These two algorithms are useful to create a set

W' of n-bit long words distant of d (in practice, the MVA finds more words than the GA, but it asks too much time for only a small improvement of the compression capacity, as shown in the tables of Figs.6 and 8, which compare the best code structures obtained for the 26-symbol English source with different values of d_{free}, respectively for the GA and for the MVA).

The second phase of the algorithm, corresponding to the elements referenced 21 to 24 (21+22 = operation A0 ; 23+24 = operation A2), consists in storing (step 21) in a set called W all the possible L_1 - tuples distant of d_{min} from each codeword in C. If that set W of all the words satisfying the minimum diverging distance to the current code is not empty (reply NO to the test 22 : $|W| = 0$?), one extra bit (a "0" or a "1") is affixed at the end of all words in W (step 24), except if the maximum number of bits is exceeded (reply YES to the test 23). At the output of said step 24, this new set replacing W has twice more words than the previous W, and the length of each one is $L_1 + 1$.

The third phase of the algorithm, corresponding to the elements 31 to 35 (= operation A3), consists in deleting (step 31) all the words of set W that do not satisfy the c_{min} distance (minimum converging distance) with all the codewords of C (i.e. in storing only those which satisfy said minimum converging distance). At this point, the new set W satisfies both d_{min} and c_{min} distances with the VLEC code C. If that new set W is not empty (reply NO to the test 32 : $|W| = 0$?) one selects in W (step 33) the maximum number of words to satisfy the minimum block distance, in order to ensure that all the words of the set W are in fact distant of b_{min}. At the end of this step 33, realized with the GA or the MVA (note that in this case, the initial set used for the GA or the MVA is the current W and not a n-tuples set), the words thus obtained are added (step 34) to the code C.

If no word is found (i.e. W is empty) at the end of the step 21 (reply YES to the test 22 : $|W| = 0$?) or if the maximum number of bits is reached or exceeded (reply YES to the test 23), one enters the fourth phase of the algorithm (steps 41 to 46, illustrated in Fig.3), which is used in order to unjam the process by inserting more liberty of choice. If there are enough codewords in the last group, some of them are deleted from this said group, as described above. Such deletions permit to reduce the distance constraint and allow to find more codewords than before. As a matter of fact, the classical Heuristic method thus described begins with the maximum of codewords with the short length, maps them with the high probability symbols and tries to obtain a good compression rate, but sometimes the size of the small lengths sets are incompatible with the required number of codewords s. In this optic, easing a few codewords provides more freedom degrees and allows to reach a position where the initial requirements on distance and number of symbols for the code can be met. This deletion process is repeated until it remains a maximum of one codeword for each length. If W is empty at the end of the step

31 (reply YES to the test 32 : $|W| = 0$?), the steps 23, 24, 31, 32 are repeated. If the required number of codewords has not been reached (reply NO to the test 35 provided at the end of this third phase), the steps 21 to 24 and 31 to 35 must be repeated until said steps find that either there is no further possibility to continue or the required number of codewords is reached.

If said required number of codewords has been reached (i.e. the number of codewords of C is equal to or greater than s (reply YES to the test 35), the structure of the VLEC code thus obtained is used in a fifth part, including the steps 51 to 56 (illustrated in Fig.4), in order to calculate the average length AL by weighting each codeword length with the probability of the source. If said average length AL of this VLEC code is lower than the minimized value of AL (= AL_{min}), this AL becomes the AL_{min} and the code structure is kept in memory 51.

To continue this search of the best VLEC code, it is necessary to avoid keeping the same structure, which would lead to a loop in the algorithm, the last group of the current code is deleted (steps 52, 53) and some codewords (half the amount for the GVA ; the "best" one for the MVA) of the previous group are deleted (step 55), in order to re-loop (step 56) the algorithm at the beginning of the step 21 and find different VLEC structures (the number of deleted codewords depends on which method is used for selecting the words : if the GA method is used and one wants to obtain a linear code, it is necessary to delete half of the codewords, while with the MVA method only one codeword, the best one, is deleted, i.e. the one that allows to find the more codewords in the next group).

However, the Heuristic method thus described often considers very unlikely code structures or proceeds with such a care (in order not to miss anything) that a great complexity is observed in the implementation of said method, which moreover is rather time consuming and can thus become prohibitive.

SUMMARY OF THE INVENTION

It is therefore an object of the invention to propose an improved construction method with which it is possible to gain in complexity by avoiding these drawbacks.

To this end, the invention relates to a method of building a variable length error code, said method comprising the steps of :

(1) initializating (phase 0) the needed parameters : minimum and maximum length of codewords L_1 and L_{max} respectively, free distance d_{free} between each codeword (said distance d_{free} being for a VLEC code C the minimum Hamming distance in the set of all arbitrary extended codes), required number of codewords s ;

(2) generating (phase 1) a fixed length code C of length L_1 and minimal distance b_{min}, with $b_{min} = \min \{b_k ; k = 1, 2, ..., R\}$, b_k = the distance associated to the codeword length L_k of code C and defined as the minimum Hamming distance between all

codewords of C with length L_k, and R = the number of different codeword lengths in C, said generating step creating a set W of n-bit long words distant of d ;

(3) storing (phase 2) in the set W all the possible L_1 – tuples distant of d_{min} from the codewords of C (said distance d_{min} for a VLEC code C being the minimum value of all the diverging distances between all possible couples of different-length codewords of C), and, if said set W is not empty, affixing at the end of all words one extra bit, said storing step replacing the set W by a new one having twice more words than the previous one and the length of each one of these words being $L_1 + 1$;

(4) deleting (phase 3) all the words of the set W that do not satisfy the c_{min} distance with all codewords of C, said distance c_{min} being the minimum converging distance of the code C ;

(5) in the case where no word is found or the maximum number of bits is reached, reducing (phase A1) the constraint of distance for finding more words ;

(6) controlling that all words of the set W are distant of b_{min}, the found words being then added to the code C ;

(7) if the required number of codewords has not been reached, repeating the steps (1) to (6) until the method finds either no further possibility to continue or the required number of codewords ;

(8) if the number of codewords of C is greater than s, calculating (phase A4), on the basis of the structure of the VLEC code, the average length AL obtained by weighting each codeword length with the probability of the source, said AL becoming the AL_{min}, if it is lower than AL_{min}, with AL_{min} = the minimum value of AL, and the corresponding code structure being kept in memory ;

said building method being moreover such that at most one bit is added at the end of each word of the set W.

It is also an object of the invention to propose a device for carrying out said construction method.

To this end, the invention relates to a device for carrying out such a variable length error code building method.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will now be described, by way of example, with reference to the accompanying drawings in which :

- Fig.1 depicts a conventional communication channel ;
- Figs. 2 to 4 are the three parts of a single flowchart illustrating the main steps of a conventional method used for building a VLEC code, called Heuristic method ;
- Fig.5 illustrates an algorithm (called greedy algorithm, or GA) used for the initialization of the method of Figs. 2 to 4, and Fig.6 is a table giving various VLEC codes for a source constructed with the Heuristic construction using said algorithm of Fig.5 ;

- Fig.7 illustrates another algorithm (called majority voting algorithm, or MVA) used for the initialization of the method of Figs. 2 to 4, and Fig.8 is another table giving various VLEC codes for a source constructed with the Heuristic construction using said algorithm of Fig.7 ;

- Figs. 9 and 10 are the two parts of a single flowchart illustrating an implementation of the method according to the invention ;

- Fig.11 is another table giving various VLEC codes for the same 26 symbol English source as considered in the tables of Figs.6 and 8 and using the GAS ;

- Fig.12 is another table giving various VLEC codes for the same source as in Fig.11 and using both the GAS previously mentioned and the building method according to the invention.

DETAILED DESCRIPTION OF THE INVENTION

Simulations show that, with the classical Heuristic method, almost none of the obtained best codes has a hole, i.e. a length jump in its structure length. It is therefore proposed, according to the invention, to consider that most good codes do not have jump of length and therefore to reduce accordingly the set of examined VLEC codes (which consequently reduces the simulation time and the complexity of implementation of the method, without modifying much the AL). Following this hypothesis, the method is modified by avoiding to add more than one bit at the end of each word of the set W.

The corresponding implementation (improved Heuristic construction method, with no hole optimization) is illustrated in Figs.9 and 10, which show the two parts of a flowchart corresponding to said improved method (the elements that are identical to the ones observed in Figs.2 to 4 being designated with the same references).

The main differences with the flowchart of Figs.2 to 4 are the following ones :

(1) first, parts that, with respect to the classical Heuristic technique, are useless for the implementation of the improved method have been cancelled :

(a) if W is empty at the end of the step 31 (reply YES to the test 32 : $|W| = 0$?), the next phase is now (see Fig.9) not the repetition of the steps (23, 24, 31, 32), but the establishment (in place of said repetition) of a direct connection 91 towards the input of the circuit carrying out the operation 55 (deletion of some codewords, or of the best one, before a repetition of the steps 21 to 24 and 31 to 35), said operation 55 being then, as previously, followed by the operations 21 and following.

(b) the fourth phase of the method is now reduced to one step, the operation 41, which is the test "Number of codewords in last group = 1 ?" . If the reply is NO, a direct link is established with the input of the step 55 (connection 91), in view of carrying out said operation 55, and then the operations 21 and following. If the reply is YES, a connection 92 is established with the input of the set of operations 52 to 54.

The results obtained with the present solution (called "noHole optimization method") are presented in the table of Fig.11 for the 26 symbol English source when using the GAS method for selecting codewords. It can be seen, when comparing with results presented in Fig.12, that although the result is not completely optimal for $d_{free} = 3$ (the code structure has a hole at length L = 11), the AL rise is really acceptable when one considers that there is both strictly no degradation for the other d_{free} values and a gain of time between 2,5 and 4. The same remarks can be applied when comparing the present solution with the ones obtained in Fig.7, where the MVA complexity effect is clear. Similarly, applying the noHole optimisation with the GA method for selecting codewords leads to a time gain at the only expense of a slight AL rise for d_{free}=3. Finally, Fig.5 shows on the other hand that the current solution offers better AL for an acceptable gain of time, the noHole optimisation compensating almost entirely the complexity induced by the GAS.

CLAIMS :

1. A method of building a variable length error code, said method comprising the steps of :

(1) initializating (phase 0) the needed parameters : minimum and maximum length of codewords L_1 and L_{max} respectively, free distance d_{free} between each codeword (said distance d_{free} being for a VLEC code C the minimum Hamming distance in the set of all arbitrary extended codes), required number of codewords s ;

(2) generating (phase 1) a fixed length code C of length L_1 and minimal distance b_{min}, with $b_{min} = \min \{b_k ; k = 1, 2, ..., R\}$, b_k = the distance associated to the codeword length L_k of code C and defined as the minimum Hamming distance between all codewords of C with length L_k, and R = the number of different codeword lengths in C, said generating step creating a set W of n-bit long words distant of d ;

(3) storing (phase 2) in the set W all the possible L_1 – tuples distant of d_{min} from the codewords of C (said distance d_{min} for a VLEC code C being the minimum value of all the diverging distances between all possible couples of different-length codewords of C), and, if said set W is not empty, affixing at the end of all words one extra bit, said storing step replacing the set W by a new one having twice more words than the previous one and the length of each one of these words being $L_1 + 1$;

(4) deleting (phase 3) all the words of the set W that do not satisfy the c_{min} distance with all codewords of C, said distance c_{min} being the minimum converging distance of the code C ;

(5) in the case where no word is found or the maximum number of bits is reached, reducing (phase A1) the constraint of distance for finding more words ;

(6) controlling that all words of the set W are distant of b_{min}, the found words being then added to the code C ;

(7) if the required number of codewords has not been reached, repeating the steps (1) to (6) until the method finds either no further possibility to continue or the required number of codewords ;

(8) if the number of codewords of C is greater than s, calculating (phase A4), on the basis of the structure of the VLEC code, the average length AL obtained by weighting each codeword length with the probability of the source, said AL becoming the AL_{min} if it is lower than AL_{min}, with AL_{min} = the minimum value of AL, and the corresponding code structure being kept in memory ;

said building method being moreover such that at most one bit is added at the end of each word of the set W.

2. A device for carrying out a variable length error code building method according to claim 1.

Abstract

The invention relates to a variable-length error-correcting (VLEC) code technique, in which the main steps are : defining all the needed parameters, generating a code having a fixed length L1, storing in a set W thus obtained all the possible L1-tuples distant of the minimum diverging distance d[min] from the codewords (one extra-bit being affixed at the end of all words if the new set W thus obtained is not empty), deleting all words of W that do not satisfy a distance criterion with all codewords, and verifying that all words of the final set W satisfy another distance criterion. Assuming that most good codes do not have jump of length, it is then proposed, according to the invention, to reduce the set of examined VLEC codes. Following this hypothesis, a new construction method, called no hole optimization, is defined, in which, it is avoided to add more than one bit at the end of each word of the set W. The new algorithm does not consider very unlikely code structures and thus allows to gain in complexity.

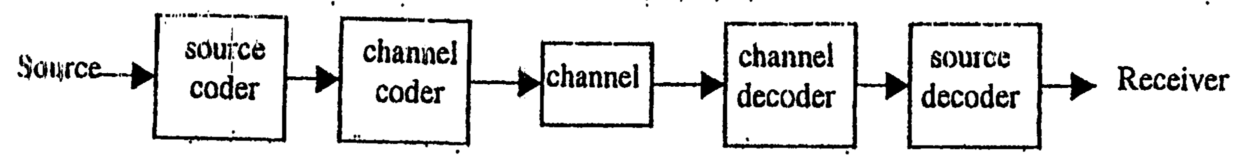


FIG. 1

d_{free}	AL	Time	Code Structure
3	6,3494	42 s	(1@4,-; 1@5,-; 5@6,3; 6@7,3; 4@8,3; 4@9,3; 1@10,-; 1@12,-; 2@13,-; 1@14,-; 2,1)
5	8,5061	3 min	(1@6,-; 2@7,5; 3@8,5; 4@9,5 ;4@10,5; 5@11,5; 5@12,5; 2@13,5; 3,2)
7	10,79	13 min	(1@8,-; 2@9,7; 2@10,7; 3@11,7; 4@12,7; 4@13,7; 6@14,7; 4@15,7; 4,3)

FIG. 11

d_{free}	AL	Time	Code Structure
3	6,3530	17 s	(1@4,-; 1@5,-; 5@6,3; 6@7,3; 4@8,3; 3@9,3; 2@10,4; 2@11,3; 1@12,-; 1@13,-; 2,1)
5	8,5061	45 s	(1@6,-; 2@7,5; 3@8,5; 4@9,5 ;4@10,5; 5@11,5; 5@12,5; 2@13,5; 3,2)
7	10,79	3 min	(1@8,-; 2@9,7; 2@10,7; 3@11,7; 4@12,7; 4@13,7; 6@14,7; 4@15,7; 4,3)

FIG. 12

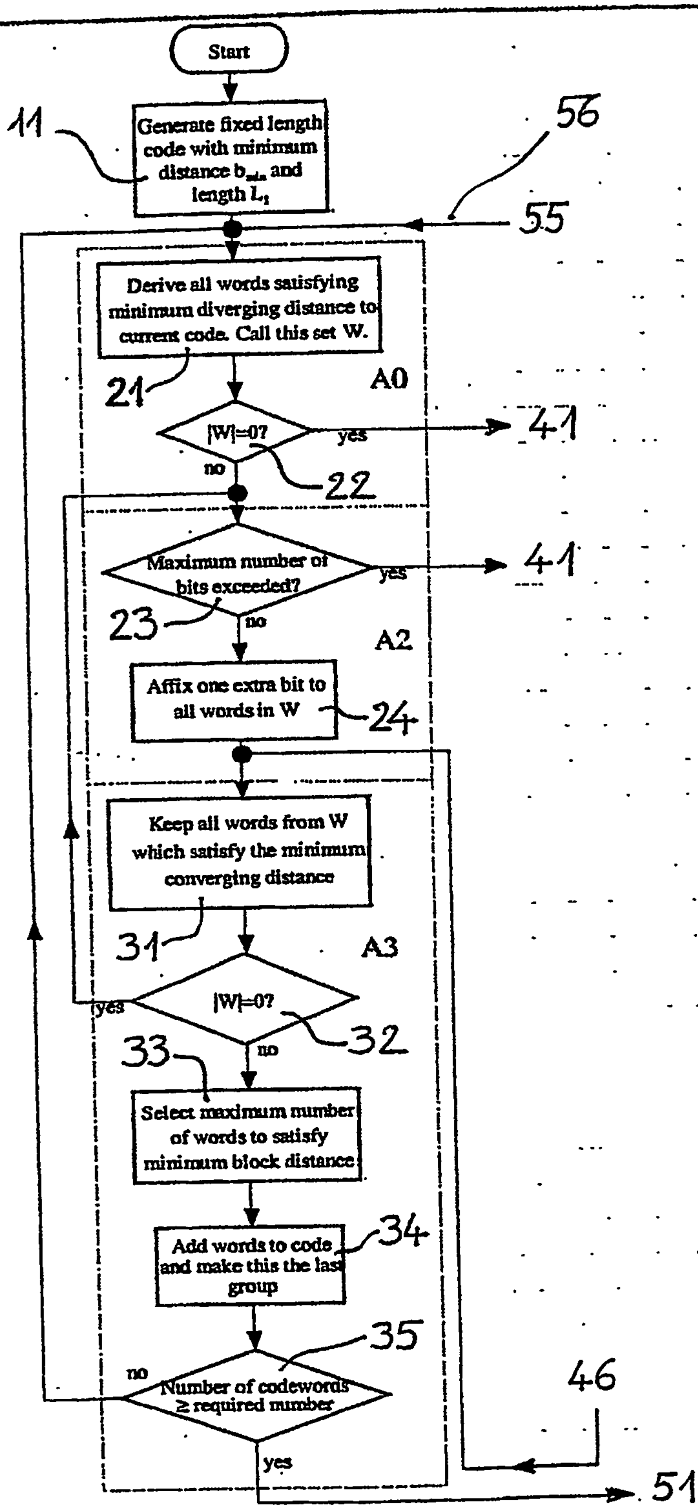

Start
11
Generate fixed length code with minimum distance b_{min} and length L_1
56
55
Derive all words satisfying minimum diverging distance to current code. Call this set W.
21
A0
|W|=0?
yes
41
no
22
Maximum number of bits exceeded?
yes
41
23
no
A2
Affix one extra bit to all words in W
24
Keep all words from W which satisfy the minimum converging distance
31
A3
yes
|W|=0?
32
33
no
Select maximum number of words to satisfy minimum block distance
Add words to code and make this the last group
34
35
no
Number of codewords ≥ required number
46
yes
51

Fig.2

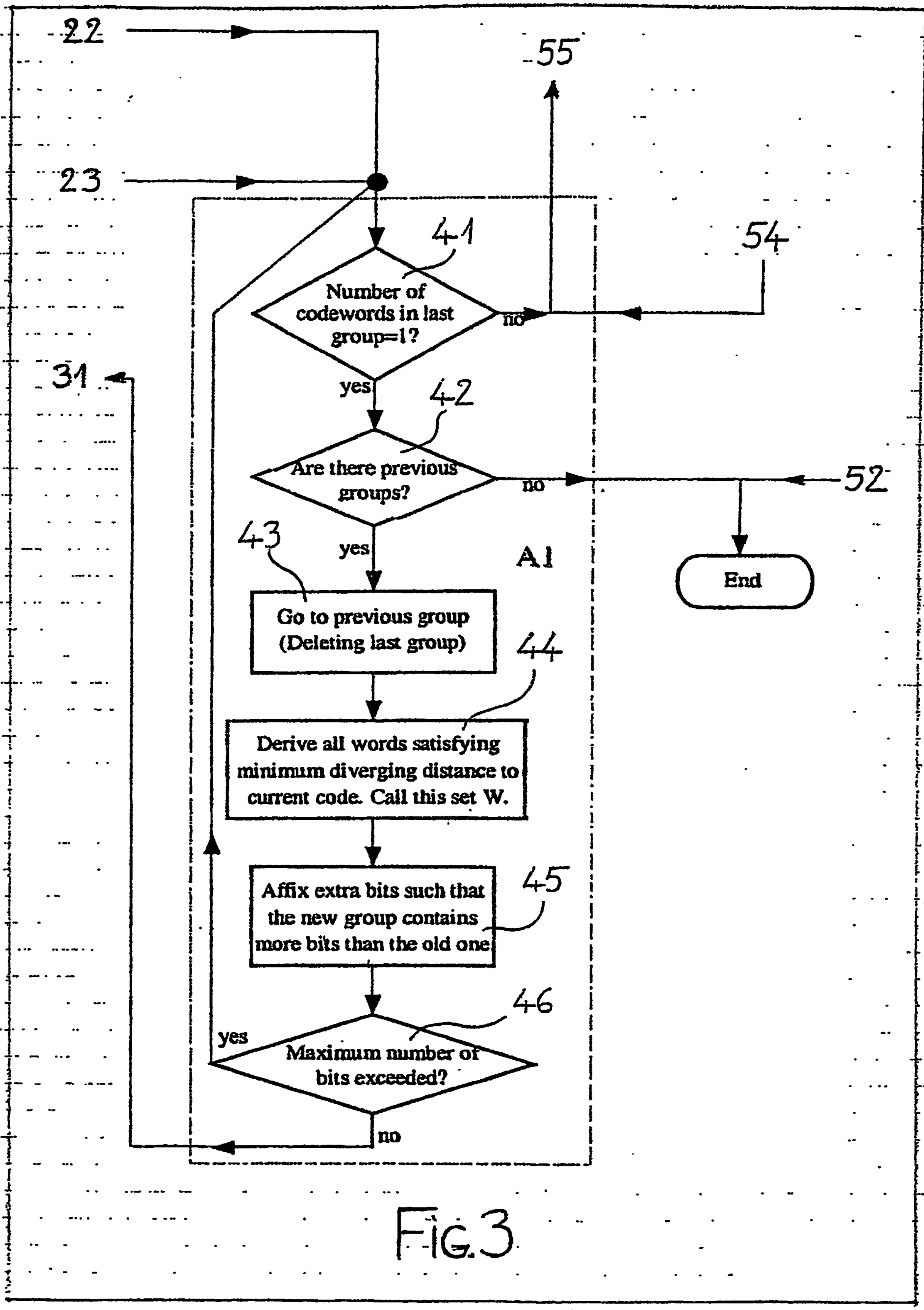
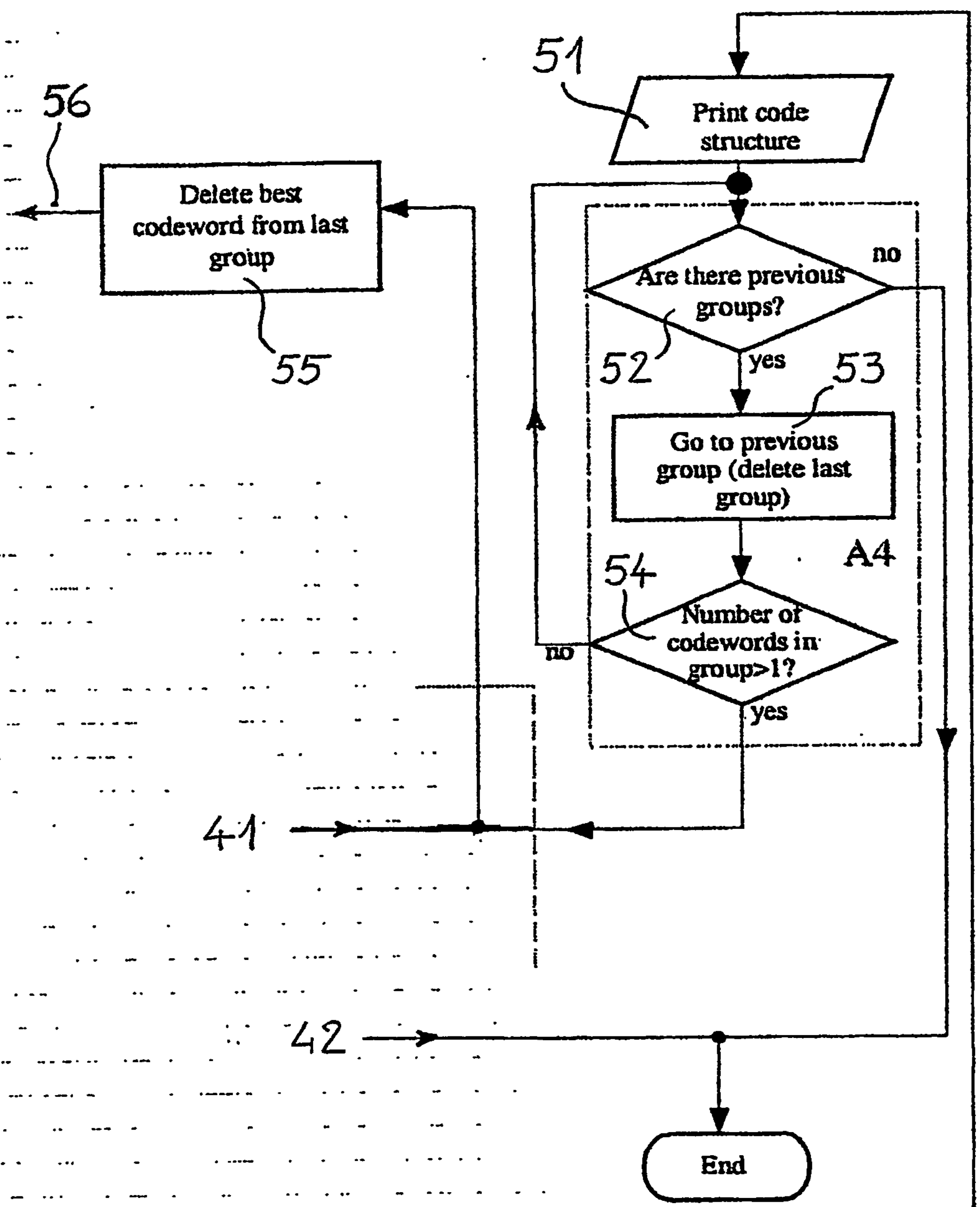

22
23
55
54
41
Number of codewords in last group=1?
no
yes
42
Are there previous groups?
no
52
End
43
yes
A1
Go to previous group (Deleting last group)
44
Derive all words satisfying minimum diverging distance to current code. Call this set W.
45
Affix extra bits such that the new group contains more bits than the old one
46
yes
Maximum number of bits exceeded?
no
31

Fig.3



56
Delete best codeword from last group
55
51
Print code structure
Are there previous groups?
no
yes
52
53
Go to previous group (delete last group)
A4
54
Number of codewords in group>1?
no
yes
41
42
End

Fig.4

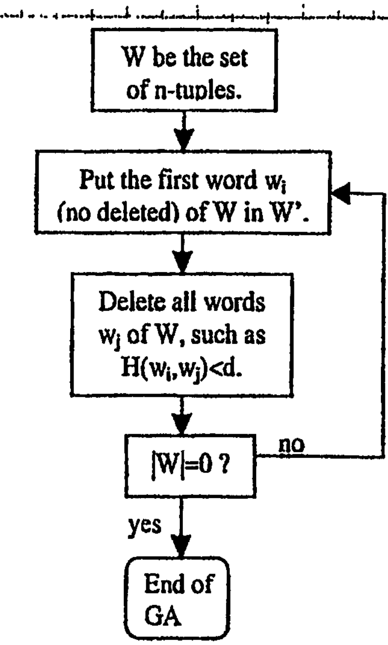


FIG. 5

d_{free}	AL	Time	Code Structure
3	6,4946	4 s	(1@4,-; 1@5,-; 3@6,3; 8@7,3; 4@8,3; 3@9,3; 1@10,-; 2@11,4; 1@12,-; 1@13,-; 1@14,-; 2,1)
5	8,5061	15 s	(1@6,-; 2@7,5; 3@8,5; 4@9,5 ;4@10,5; 5@11,5; 5@12,5; 2@13,5; 3,2)
7	10,79	2 min	(1@8,-; 2@9,7; 2@10,7; 3@11,7; 4@12,7; 4@13,7; 6@14,7; 4@15,7; 4,3)

FIG. 6

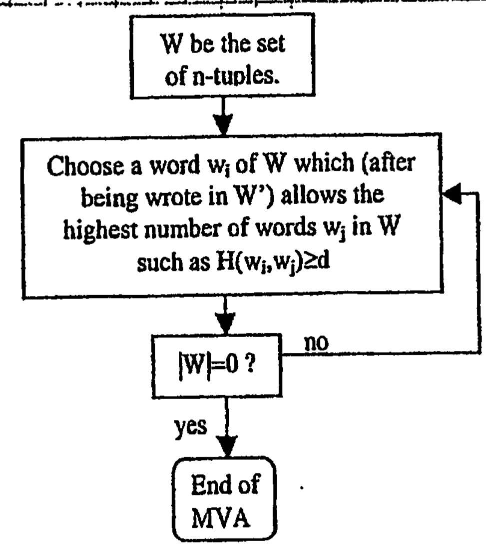


FIG.7

d_{free}	AL	Time	Code Structure
3	6,3038	50 s	(1@4,-; 2@5,4; 4@6,3; 4@7,3; 6@8,3; 5@9,3; 2@10,4; 2@11,3; 2,1)
5	8,4752	16 min	(1@6,-; 1@7,-; 4@8,5; 5@9,5; 5@10,5; 5@11,5; 5@12,5; 3,2)
7	10,7385	14 h	(1@7,-; 1@8,-; 1@9,-; 1@10,-; 3@11,7; 4@12,7; 5@13,7; 5@14,7; 5@15,7; 4,3)

FIG.8

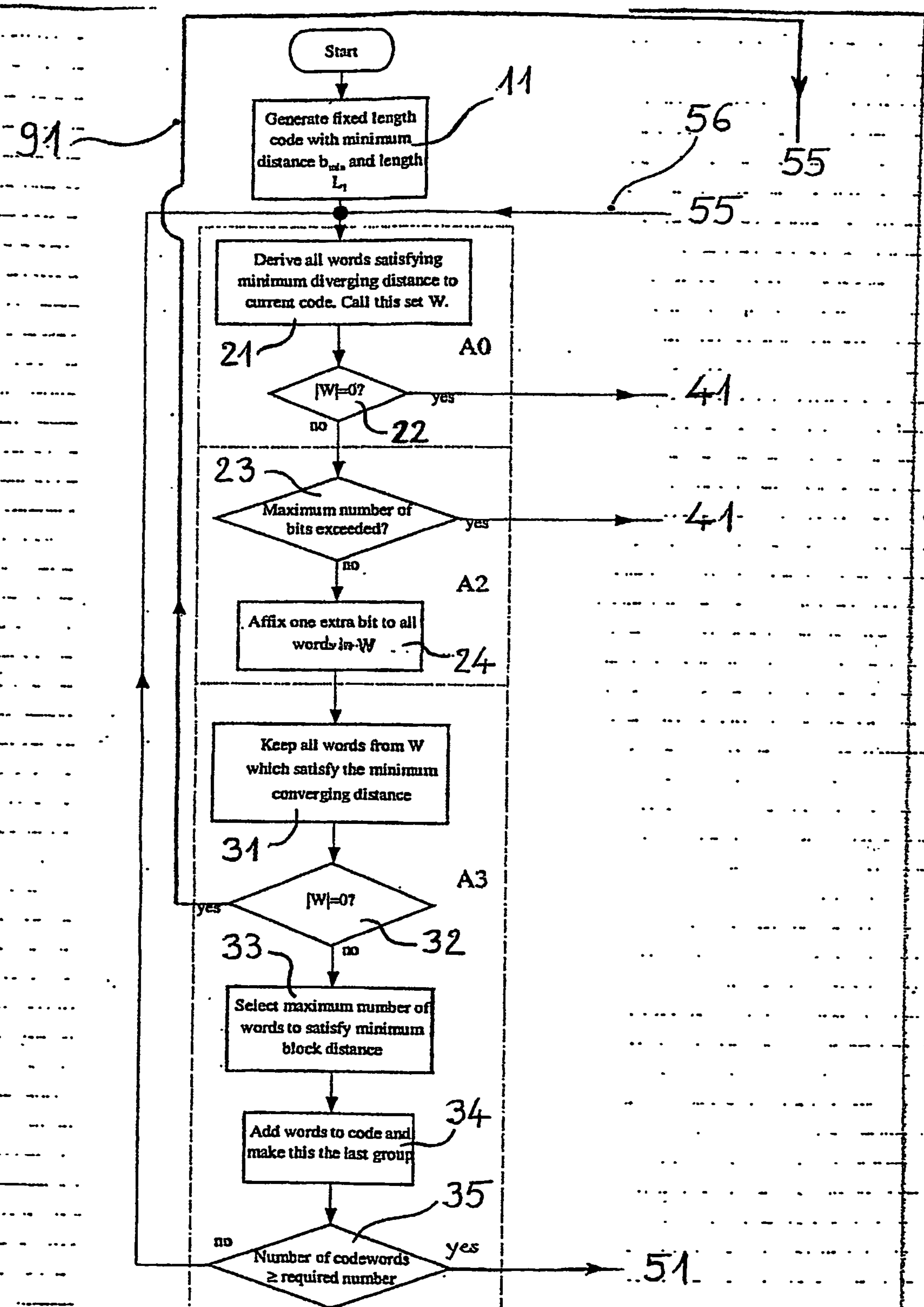

Start
Generate fixed length code with minimum distance b_{min} and length L_1
11
91
56
55
55
Derive all words satisfying minimum diverging distance to current code. Call this set W.
21
A0
|W|=0?
yes
no
22
41
23
Maximum number of bits exceeded?
yes
no
41
A2
Affix one extra bit to all words in W
24
Keep all words from W which satisfy the minimum converging distance
31
A3
|W|=0?
yes
no
32
33
Select maximum number of words to satisfy minimum block distance
34
Add words to code and make this the last group
35
Number of codewords ≥ required number
no
yes
51

Fig.9

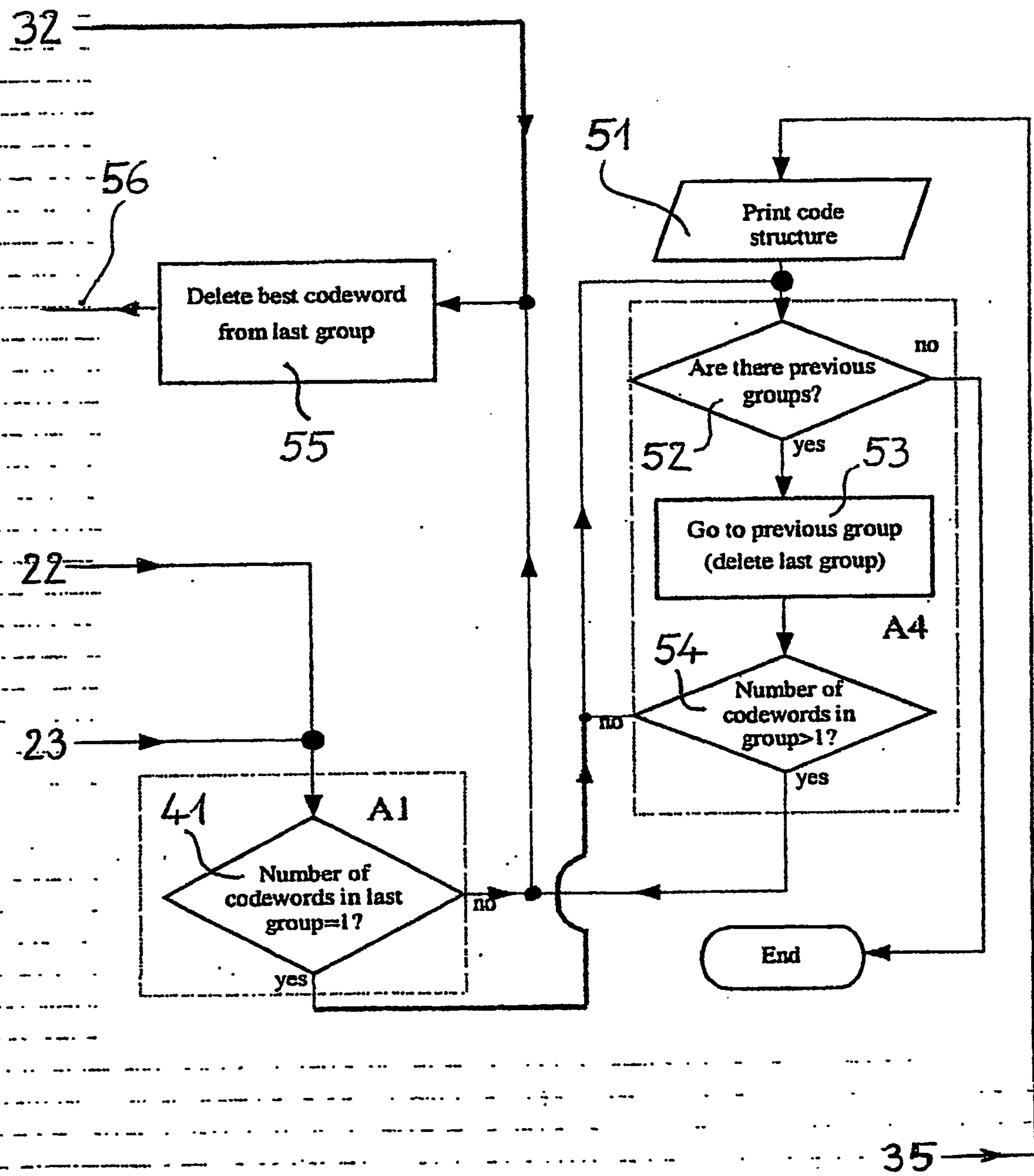

32
56
Delete best codeword from last group
55
51
Print code structure
Are there previous groups?
no
yes
52
53
Go to previous group (delete last group)
A4
54
Number of codewords in group>1?
no
yes
22
23
41
A1
Number of codewords in last group=1?
no
yes
End
35

Fig.10